UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/929,742	08/13/2001	Falko Tesch	30014200-1012	9030

58328   7590   08/08/2006

SONNENSCHEIN NATH & ROSENTHAL LLP
FOR SUN MICROSYSTEMS
P.O. BOX 061080
WACKER DRIVE STATION, SEARS TOWER
CHICAGO, IL 60606-1080

EXAMINER
NGUYEN BA, PAUL H

ART UNIT	PAPER NUMBER
2176	

DATE MAILED: 08/08/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

	Application No.	Applicant(s)
Office Action Summary	09/929,742	TESCH ET AL.
	Examiner	Art Unit	
	Paul Nguyen-Ba	2176	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on <u>*14 July 2006*</u>.

2a) ☐ This action is **FINAL**.   2b) ☒ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) *1-28,31-33 and 36-38* is/are pending in the application.

     4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) *1-28, 31-33, 36-38* is/are rejected.

7) ☐ Claim(s) _____ is/are objected to.

8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.

10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.

     Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

     Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

     a) ☐ All   b) ☐ Some *   c) ☐ None of:

         1. ☐ Certified copies of the priority documents have been received.

         2. ☐ Certified copies of the priority documents have been received in Application No. _____.

         3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

     * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1.   A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 7/14/2006 has been entered.

2.   Claims 1-28, 31-33, 36-38 are currently pending. Claims 1, 7, 8, 9, 10, 11, 17, 18, 19, 20, 25, 26, 31, 36, 37, and 38 are independent claims.

Claim Objections

3.   **Claims 1, 8, 11, 17, 18, 25, 26, 31, 36, 37, and 38 are objected to because of the following informalities:**

> ➤ Typographical Error – The "**date** processing system" should be changed to the "**data** processing system". Appropriate correction is required.

Claim Rejections - 35 USC § 103

4.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set
> forth in section 102 of this title, if the differences between the subject matter sought to be patented and
> the prior art are such that the subject matter as a whole would have been obvious at the time the
> invention was made to a person having ordinary skill in the art to which said subject matter pertains.
> Patentability shall not be negatived by the manner in which the invention was made.

5.   Claims 1-5, 7-15, 17-19, 25-28, and 31-33 are rejected under 35 U.S.C. 103(a)

as being unpatentable over Oracle Forms ® Advanced Techniques ("Oracle"), Ch.

10, pgs. 1-18, © 1996 Oracle Corporation (*available at*

http://mates.ms.mff.cuni.cz/oracle/doc/forms45/at/ch10.htm), in view of Austin,

U.S. Patent Application No. 2003/0037119.

Regarding independent claims 1 and 7, Oracle teaches a method in a data

processing system for processing a document containing an embedded object having a

first format corresponding to a first program (i.e., OLE) (see pgs. 2-3), the method

comprising the steps of:

> ➤ *initiating loading of the document into a memory of the data processing*
> *system* (This feature is deemed inherent in data processing systems. Without the
> loading of the document into memory, the instant invention would cease to
> function or operate because no data would be available to process); *and*

> ➤ while the document is being loaded into the memory, *identifying*
> *the embedded object contained in the document* (see pg. 2 *et seq.* OLE

provides the capability to integrate and identify objects from many

identifiable application programs into a single document),

> while the document is being loaded into the memory,

automatically determining whether the first program is an unavailable

program (see pg. 17, heading: Converting OLE Objects – 1st paragraph.

OLE object conversion is used for editing OLE objects when the OLE

server application that originated an OLE object is **not available**. A

program is automatically determined to be unavailable when an

embedded object cannot be run or displayed in a document because its

designated executable program is not available);

> *when it is determined that the first program is an unavailable*

program, converting the embedded object into a second format different

from the first format that is suitable for use with a second program that is

available on the data processing system (see pg. 17 and 18, headings:

Converting OLE Objects and Converting Embedded Objects. The

"Convert To" command permanently alters the format of the object to the

selected type for *automatic identification* of the selected type);

> *receiving an indication of a third format from a user* (see pg. 18,

step 3. The convert dialog shows the current object type and the

conversion possibilities);

> *converting the embedded object into the third format* (see pg. 18,

step 4);

> *storing the embedded object in the third format* (pg. 18, step 5).

Oracle does not explicitly teach automatically converting the embedded object into a second and third formats *while the document is being loaded into memory*.

However, Austin teaches a graphical program wherein during program execution (i.e., "while the document is being loaded into memory") the data access node object is operable to automatically convert the data object to a format understandable by the graphical program (see [0121-0123]). The format may be generic or indicated from a user (see [0121-0123]).

Since Oracle and Austin are both from the same field of endeavor, the motivational purpose of freeing authors from the administrative burdens associated with maintaining different versions of an object and having to manually convert objects to executable formats as disclosed by Austin would have been recognized in the pertinent art of Oracle. It would have been obvious at the time the invention was made to a person having ordinary skill in the art to modify the teaching of Oracle with the teachings of Austin to include automatically converting the embedded object into a second and third formats while the document is being loaded into memory.

Independent claims 9, 10, 11, 17, 19, and 25 incorporate substantially similar subject matter as independent claim 7, and are rejected along the same rationale.

Regarding claims 2-5, 12, 14, 15, Oracle, in view of Austin, teach:

> ➢ receiving an indication of the second and third format from a user,

determining and displaying the associated formats of the available programs to

the user (see Oracle - pg. 18, step 3. The convert dialog shows the current

object type and the conversion possibilities),

> ➢ converting the embedded object into a third format and storing the

embedded object (see Oracle - pg. 18, step 4 and step 5), and

automatically identifying a second format (see Oracle - pg. 17 and 18, headings:

Converting OLE Objects and Converting Embedded Objects. The "Convert To"

command permanently alters the format of the object to the selected type for

automatic identification of the selected type).

Regarding independent claim 8, Oracle teaches a method in a data processing

system containing a plurality of programs, each with an associated format, the data

processing system for processing a document containing an embedded object having

an originating format corresponding to an originating program (i.e. OLE) (see pgs. 2-3),

the method comprising the steps of:

> ➢ *initiating loading of the document into a memory of the data processing*

system (This feature is deemed inherent in data processing systems. Without the

loading of the document into memory, the instant invention would cease to

function or operate because no data would be available to process); *and*

➢ while the document is being loaded into the memory, *identifying the embedded object contained in the document* (see pgs. 2-18. OLE provides the capability to integrate and identify objects from many identifiable application programs into a single document),

➢ *automatically determining whether the originating program is unavailable* (see pg. 17, heading: Converting OLE Objects – 1ˢᵗ paragraph → OLE object conversion is used for editing OLE objects when the OLE server application that originated an OLE object is **not available**. A program is automatically determined to be unavailable when an embedded object cannot be run or displayed in a document because its designated executable program is not available);

➢ *when it is determined that the originating program is unavailable, determining which of the plurality of programs are available on the data processing system* (see pg. 18, step 3; see also pg. 17 and 18, headings: Converting OLE Objects and Converting Embedded Objects. The "Convert To" command permanently alters the format of the object to the selected type for *automatic identification* of the selected type.),

➢ *displaying the associated formats of the available programs to a user* (see Figure on pg. 17), and

➢ *receiving an indication of a selected one of the displayed formats from the user* (see pg. 18, step 3. The convert dialog shows the current object type and the conversion possibilities); and

> ➢ *converting the embedded object into the selected format* (see pg.

18, step 4).

Oracle does not explicitly teach automatically converting the embedded object

into a second format *while the document is being loaded into memory*.

However, Austin teaches a graphical program wherein during program execution

(i.e., "while the document is being loaded into memory") the data access node object is

operable to automatically convert the data object to a format understandable by the

graphical program (see [0121-0123]). The format may be generic or indicated from a

user (see [0121-0123]).

Since Oracle and Austin are both from the same field of endeavor, the

motivational purpose of freeing authors from the administrative burdens associated with

maintaining different versions of an object and having to manually convert objects to

executable formats as disclosed by Austin would have been recognized in the pertinent

art of Oracle. It would have been obvious at the time the invention was made to a

person having ordinary skill in the art to modify the teaching of Oracle with the teachings

of Austin to include automatically converting the embedded object into a second and

third formats while the document is being loaded into memory.

Regarding claim 13, Oracle, in view of Austin, teach determining which of the

plurality of programs are available on the data processing system (see Oracle - pg. 18,

step 3) and displaying the associated formats of the available programs to a user (see

Figure on pg. 17).

Independent claims 18, 26, and 31 incorporate substantially similar subject

matter as independent claim 8, and are rejected along the same rationale.

Regarding claims 27 and 32, Oracle, in view of Austin, teach receiving the

indication from a user and retrieving the indication from storage (see Oracle - pg. 18,

step 4 and step 5).

Regarding claims 28 and 33, Oracle, in view of Austin, teach *retrieving the*

indication from storage (see Oracle - pg. 17 and 18, specifically step 5 → the "Convert

To" command permanently alters the format of the object to the selected type for

automatic identification of the selected type and is stored and recalled from storage).

6.   Claims 6, 16, 20, 21, 22-24 are rejected under 35 U.S.C. 103(a) as being

unpatentable over Oracle Forms ® Advanced Techniques ("Oracle"), Ch. 10, pgs.

1-18, © 1996 Oracle Corporation, in view of Austin, U.S. Patent Application No.

2003/0037119, in further view of Francis et al. ("Francis"), U.S. Patent No.

6,182,092,

Regarding claims 6 and 16, Oracle, in view of Austin, teach the method and computer readable medium of independent claims 1 and 11, but does not specifically teach converting the embedded object into an intermediate format.

However, Francis teaches converting OLE documents and objects into an intermediate format as a preprocessing step (see Fig. 6 and col. 14, lines 24-40) for the purpose instantiating the output, and hence, smoothing the transition between different formats.

It would have been obvious at the time the invention was made to a person having ordinary skill in the art to modify the teaching of Oracle, in view of Austin, with the teachings of Francis to include converting the embedded object into an intermediate format for the purpose instantiating the output, and hence, smoothing the transition between different formats.

Regarding independent claim 20, Oracle, in view of Austin, teach the method for processing a document containing an embedded object as discussed in independent claim 1 above, but does not specifically teach a first or second identifier wherein the second identifier can replace the first identifier.

However, Francis teaches the use of identifiers to identify objects of a format embeddable in the document (see col. 2 lines 52-53 and col. 4 39-67 *et seq.*) for the purpose of associating and identifying different embedded objects in a document.

Furthermore, it is well known to those of ordinary skill in the art that a first

identifier can be replaced by a second identifier for the purpose of converting a first

object format into a second object format.

Regarding claim 21, Oracle, in view of Austin, does not specifically teach a first

or second identifier wherein the second identifier can replace the first identifier.

However, Francis teaches the use of identifiers to identify objects of a format

embeddable in the document (see col. 2 lines 52-53 and col. 4 39-67 *et seq.*) for the

purpose of associating and identifying different embedded objects in a document.

Furthermore, it is well known to those of ordinary skill in the art that a first

identifier can be replaced by a second identifier for the purpose of converting a first

object format into a second object format.

Regarding claims 22-24, please refer to the rationale relied upon to reject

independent claim 1 above.

7. **Claims 36-38 are rejected under 35 U.S.C. 103(a) as being unpatentable**

over Oracle Forms ® Advanced Techniques ("Oracle"), Ch. 10, pgs. 1-18, © 1996

Oracle Corporation, in view of Laverty et al. ("Laverty"), U.S. Patent No. 6,396,593,

in further view of Austin, U.S. Patent Application No. 2003/0037119.

Independent Claims 36-38

Oracle, in view of Austin, teach the method, system, and computer-readable

medium with respect to independent claim 1 as discussed above, but does not

specifically teach selecting a user selectable setting comprising at least a first setting for

performing the step of converting while the document is being loaded into memory and

a second setting for performing the step of converting upon selection of the document

for editing.

However, Laverty teaches user selectable conversion settings (see col. 6 lines

38-40) for the motivational purpose of allowing the human user to intervene, oversee,

and drive all steps in the conversion process.

It would have been obvious at the time the invention was made to a person

having ordinary skill in the art to modify the teaching of Oracle with the teachings of

Laverty to include a choice of settings for performing the step of converting at various

points of the conversion process for the motivational purpose of allowing the human

user to intervene, oversee, and drive all steps in the conversion process.

Oracle does not explicitly teach automatically converting the embedded object

into a second format *while the document is being loaded into memory*.

However, Austin teaches a graphical program wherein during program execution

(i.e., "while the document is being loaded into memory") the data access node object is

operable to automatically convert the data object to a format understandable by the

graphical program (see [0121-0123]). The format may be generic or indicated from a

user (see [0121-0123]).

Since Oracle and Austin are both from the same field of endeavor, the

motivational purpose of freeing authors from the administrative burdens associated with

maintaining different versions of an object and having to manually convert objects to

executable formats as disclosed by Austin would have been recognized in the pertinent

art of Oracle. It would have been obvious at the time the invention was made to a

person having ordinary skill in the art to modify the teaching of Oracle with the teachings

of Austin to include automatically converting the embedded object into a second and

third formats while the document is being loaded into memory.

Response to Arguments

8.   Applicant's arguments with respect to claims filed on 7/14/2006 have been

considered but are moot in view of the new ground(s) of rejection.

The new ground of rejection includes the addition of the Austin reference, which

is being relied upon for teaching the limitation, "automatically converting the embedded

object into a second format *while the document is being loaded into memory*".

Applicant's arguments focus on the prior art's failure to teach this particular limitation.

One of ordinary skill in the art would have been motivated at the time of the invention to

arrive at the instant invention by combining Oracle with Austin.

Regarding the newly amended limitation "*initiating loading of the document into a*

memory of the data processing system". This feature is deemed inherent in data

processing systems. Without the loading of the document into memory, the instant

invention would cease to function or operate because no data would be available to

process.

Regarding the newly amended limitation "*identifying the embedded object*

contained in the document ". Oracle teaches this starting on page 2 of the reference.

Oracle teaches that OLE provides the capability to integrate and identify objects from

many identifiable application programs into a single document.

Conclusion

9.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Paul Nguyen-Ba whose telephone number is (571) 272-

4094. The examiner can normally be reached on 11 am - 7 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Heather Herndon can be reached on (571) 272-4136. The fax phone

number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

PNB
8/2/06

Heather R. Herndon
Supervisory Patent Examiner
Technology Center 2100